Numerical modelling of the blowing phase in the production of glass containers

W. Dijkstra*, R.M.M. Mattheij*

*Department of Mathematics and Computing Science, Eindhoven University of Technology,

P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Abstract

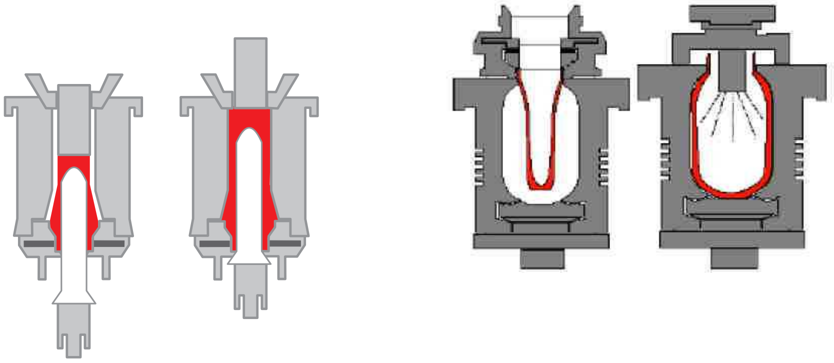
The industrial manufacturing of glass containers consists of several phases, one of which is the blowing phase. This paper describes the development of a numerical simulation tool for this phase. The hot liquid glass is modelled as a viscous fluid and its flow is governed by the Stokes equations. We use the boundary element method to solve the Stokes equations and obtain the velocity profile at the glass surface. The movement of the surface obeys an ordinary differential equation. We use the Euler forward method to perform a time integration step and update the shape of the glass surface. All calculations are performed in three dimensions. This allows us to simulate the blowing of glass containers that are not rotationally symmetric. The contact between glass and mould is modelled using a partial-slip condition. A number of simulations on model glass containers illustrates the results.

1 Introduction

The industrial production of glass products like bottles and jars consists of several phases. First glass is melted in a furnace where the glass reaches temperatures between 1200 and 1600 ^{o}C. The molten glass is then cut into *gobs*, which are transported to a forming machine.

The gob is positioned into a mould that is open from below. A plunger is pushed into the mould, shaping the glass to an intermediate form called the *parison*. This phase of the production process is called the *pressing phase* (Figure 1(a)). The parison is put into a second mould in which it is allowed to sag vertically due to gravity for a short period. When the glass almost touches the bottom of the mould, pressurized air flows into the mould from above, blowing the glass to its final shape. This phase of the production process is called the *blowing phase* (Figure 1(b)). After the blowing phase the glass is removed from the mould.

For the glass industry it is important to optimize each phase of the production process. One can think of optimizing the shape of the parison, the speed of the plunger, the sagging



(a) *Pressing phase* (b) *Blowing phase*

Figure 1: *The production of glass containers consists of a pressing phase and a blowing phase.*

time, the pressure of the air during the blowing phase, etc [1]. Experiments to tune these parameters are cumbersome, costly and time consuming. Therefore computer simulation of the various production phases can offer useful information to optimize the production.

The industrial process sketched above is a typical example of a class of problems that can be described as *blowing problems*. With this classification we mean the shaping of products in a liquid phase by a blowing mechanism. The goal of this paper is to use the Boundary Element Method (BEM) to numerically simulate such blowing problems. In particular we focus on the blowing problem for glass products.

As we are interested in the shape evolution of the surface of the glass we do not need to know the fluid flow in the interior of the glass. The BEM only computes the velocity profile at the surface of the glass, whereas many other numerical methods also need to compute the glass velocity at internal points. Hence, from this point of view, the BEM seems to be a very appropriate numerical method for this blowing problem. Throughout this paper we will investigate how effective the boundary element method is for solving the blowing problem.

We assume that the initial shape of the parison, the shape of the mould, the pressure of the pressurized air and material parameters are given. The hot liquid glass is a higly viscous fluid, and its flow is described by the Stokes equations. The BEM computes the flow at the surface of the glass solving these Stokes equations. Then we perform a time integration step to obtain the shape of the glass at the next time level. For this new shape we again compute the flow at the surface and perform another time integration step. This iterative procedure enables us to study the shape evolution of the glass during the blowing phase. The computations are performed in three dimensions. This allows us to study bottles and jars that are not rotationally symmetric, for instance due to small imperfections in the initial parison.

Numerical modelling of the production process of glass bottles and jars has been the topic of several papers. Mostly finite elements are used to solve the Stokes equations [2, 3], sometimes using a level set method to track the position of the glass surface [4]. In many cases rotationally symmetric parisons are modelled and computations are thus limited to two dimensions. To the authors' knowledge our work is the first to address the blowing problem in three dimensions using the BEM. Other problems have been handled successfully by the BEM in combination with a level set method, such as the simulation of crystal growth [5], dynamic powder consolidation of metals [6] and breaking waves over a sloping beach [7].

During the blowing phase the temperature of the glass changes due to heat exchange with the mould. The viscosity of the glass depends on the temperature in an essentially non-linear way. As the viscosity appears in the Stokes equations, the heat problem and the flow problem are coupled. In the papers mentioned above this phenomenon is studied intensively. In our paper we assume that the glass has a uniform temperature that remains constant during the whole blowing phase. This allows us to focus at the flow problem only.

Special attention has to be given to the contact problem of the glass and the mould. Most papers assume a no-slip condition at the mould. In practice this is not the case. Sometimes the mould is even covered with a lubricating substance to improve the slip of the glass. Therefore we choose to work with a partial-slip boundary condition instead of a no-slip boundary condition.

The procedure described above results in a simulation tool to study the blowing phase for glass products. We have tested the simulation tool on several bottles and jars. The results of the tests are promising and may contribute to a better understanding of the production of bottles and jars.

This paper is set up as follows. Section 2 introduces the mathematical model of the glass flow during the blowing phase. The boundary value problem that we derive in this section is solved with the boundary element method in Section 3. Section 4 shows an algorithmic approach for simulating the blowing phase. In Section 5 we present a number of tests to show the performance of the simulation tool. We conclude with a brief discussion in Section 6

2 Mathematical model

In this section we derive the mathematical model that describes the flow of a three-dimensional volume of Newtonian fluid with high viscosity.

We consider a volume of fluid in three dimensions denoted by Ω. The fluid is bounded by a closed surface S. The velocity and pressure of the fluid are denoted by $\mathbf{v}$ and p respectively. Furthermore the fluid is characterized by the dynamic viscosity η, the surface tension γ and a typical length scale L. In general the viscosity η depends on the temperature of the glass. As the temperature may be space dependent, also the viscosity may be space dependent. However, later on we show that for the blowing problem the temperature remains constant along streamlines. As a consequence we can assume that

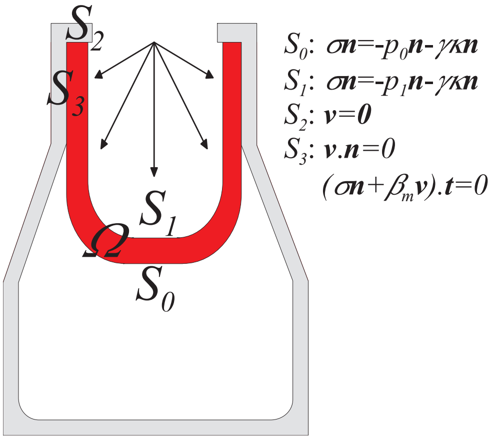


Figure 2: *The surface of the glass is divided into four parts (cross-sectional view).*

the glass has a uniform temperature distribution, and henceforth the viscosity is uniform also.

The motion of the fluid is governed by two equations. The continuity equation expresses conservation of mass, and reads

$$\nabla.\mathbf{v} = 0, \tag{1}$$

where we assumed that the density of the fluid is constant and uniform, i.e. the fluid is incompressible. The conservation of momentum is described by the Navier-Stokes equations,

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho(\nabla.\mathbf{v})\mathbf{v} = \nabla.\sigma + \rho\mathbf{g}, \tag{2}$$

where $\mathbf{g}$ is a body force (here we consider only gravitational force $\mathbf{g} = -g\mathbf{e}_z$), ρ is the density and σ is the stress tensor. For a Newtonian fluid the following constitutive equation for the stress tensor holds,

$$\sigma_{ij} := -p\delta_{ij} + \eta\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right), \tag{3}$$

with δ_{ij} the Kronecker delta. Substitution of the constitutive equation for σ into the Navier-Stokes equations yields

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho(\nabla.\mathbf{v})\mathbf{v} = -\nabla p + \eta\nabla^2\mathbf{v} + \rho\mathbf{g}. \tag{4}$$

We distinguish four types of boundary conditions, see Figure 2. At the surfaces S_0 and S_1 the normal stress is related to the prescribed pressures p_0 and p_1 onto the surface and the surface tension γ,

$$\begin{aligned} \sigma\mathbf{n} &= -p_0\mathbf{n} - \gamma\kappa\mathbf{n}, \text{ at } S_0, \\ \sigma\mathbf{n} &= -p_1\mathbf{n} - \gamma\kappa\mathbf{n}, \text{ at } S_1. \end{aligned} \tag{5}$$

The vector $\mathbf{n}$ is the outward unit normal at the surface and κ is the mean curvature at a certain point of the surface. The first term in the boundary condition accounts for the external pressure acting onto the surface. The second term accounts for the surface tension due to the curvature of the surface. In the fluid all molecules attract one another. A molecule that is in the interior of the fluid domain is attracted by all its neighbours, so the average force it experiences is equal to zero. A molecule at the surface of the fluid experiences a force inwards the fluid. For highly curved surfaces this force will be larger than for flat surfaces. The curvature of the surface is measured by the *mean curvature* κ with dimension L^{-1}. For more details about the incorporation of curvature in the boundary conditions we refer to [8].

At the surface S_2 the glass is not allowed to move and hence we set the velocity equal to zero,

$$\mathbf{v} = \mathbf{0}, \text{ at } S_2. \tag{6}$$

At the surface S_3 the fluid is in contact with a solid wall, but is allowed to slip along the wall. This means that the velocity component in the normal direction is equal to zero, i.e. the fluid cannot penetrate through the wall,

$$\mathbf{v}.\mathbf{n} = 0, \text{ at } S_3. \tag{7}$$

The velocity component in the tangential directions does not need to be zero. The most common condition is that the tangential component of the velocity is related to the normal stress by [9, 10],

$$(\sigma\mathbf{n} + \beta_m\mathbf{v}).\mathbf{t} = 0, \text{ at } S_3. \tag{8}$$

Here $\mathbf{t}$ is a vector in the tangential direction at the wall and β_m is a friction parameter. If $\beta_m \to 0$ there is no friction between fluid and wall. If $\beta_m \to \infty$ the friction between fluid and wall is too large to allow slip along the wall. It can be seen that in that case (7) together with (8) yield the no-slip condition (6).

We introduce a dimensionless pressure p', velocity $\mathbf{v}'$ and body force $\mathbf{g}'$ by

$$p' := \frac{p - p_0}{p_1 - p_0}, \quad \mathbf{v}' := \frac{\mathbf{v}}{v_c} := \frac{\eta\mathbf{v}}{(p_1 - p_0)L}, \quad \mathbf{g}' := -\mathbf{e}_z. \tag{9}$$

We also define a characteristic time scale $t_c := L/v_c$. Using these characteristic variables and using L as a characteristic length, we rewrite the Navier-Stokes equations in dimensionless form,

$$\text{Re}\left(\frac{\partial\mathbf{v}'}{\partial t'} + (\nabla'.\mathbf{v}')\mathbf{v}'\right) = -\nabla' p' + \nabla'^2\mathbf{v}' + \frac{\text{Re}}{\text{Fr}}\mathbf{g}'. \tag{10}$$

Here the differential operator ∇' denotes differentiation with respect to the dimensionless spatial coordinates. The dimensionless numbers Re and Fr are the Reynolds number and the Froude number, defined as

$$\text{Re} := \frac{\rho L v_c}{\eta}, \quad \text{Fr} := \frac{v_c^2}{gL}, \tag{11}$$

where g is the acceleration of gravity. The Reynolds number is very small for viscous fluids (10^{-3} or smaller for liquid glass) and therefore the terms at the left-hand side of the momentum balance (10) can be neglected. The ratio of the Reynolds number and Froude number does not need to be very small and we therefore we do not neglect the contribution of the gravity. Hence the Navier-Stokes equations reduce to

$$-\nabla' p' + \nabla'^2 \mathbf{v}' + \alpha \mathbf{g}', \tag{12}$$

where α is the ratio of the Reynolds number and Froude number, $\alpha = \mathrm{Re}/\mathrm{Fr}$. The equations (12) are called the Stokes equations. The dimensionless form of the momentum balance (12) together with the dimensionless form of the mass balance (1),

$$\nabla'.\mathbf{v}' = 0, \tag{13}$$

give a system of four equations that describe the flow of the fluid.

It can be verified that the dimensionless stress tensor σ' is defined as

$$\sigma' := -\frac{p_0}{p_1 - p_0}\mathcal{I} + \frac{1}{p_1 - p_0}\sigma. \tag{14}$$

We also introduce a dimensionless curvature κ' by $\kappa' = L\kappa$. Substitution of σ' and κ' into the boundary conditions at S_0 and S_1 yields

$$\begin{aligned} \sigma'\mathbf{n} &= -\beta\kappa'\mathbf{n}, \text{ at } S_0, \\ \sigma'\mathbf{n} &= -(1 + \beta\kappa')\mathbf{n}, \text{ at } S_1, \end{aligned} \tag{15}$$

where the dimensionless parameter β is defined as

$$\beta := \frac{\gamma}{(p_1 - p_0)L}. \tag{16}$$

The boundary condition at S_2 becomes

$$\mathbf{v}' = \mathbf{0}, \text{ at } S_2. \tag{17}$$

It can be verified that substitution of σ' and $\mathbf{v}'$ into the second part of the boundary condition at S_3 yields

$$-p_0\mathbf{n}.\mathbf{t} + (p_1 - p_0)(\sigma'\mathbf{n} + \frac{L\beta_m}{\eta}\mathbf{v}').\mathbf{t} = 0. \tag{18}$$

The first term cancels out since $\mathbf{n}.\mathbf{t} = 0$. We divide by $p_1 - p_0$ and and get the following boundary conditions at S_3,

$$\begin{aligned} (\sigma'\mathbf{n} + \beta'_m\mathbf{v}').\mathbf{t} &= 0, \\ \mathbf{v}'.\mathbf{n} &= 0, \text{ at } S_3, \end{aligned} \tag{19}$$

where the dimensionless friction parameter β'_m is defined as

$$\beta'_m := \frac{L\beta_m}{\eta}. \tag{20}$$

In the sequel we drop the $'$ to simplify the notation.

We introduce a modified pressure $\tilde{p}$ by [11, p. 164]

$$\tilde{p} := p + \alpha z, \tag{21}$$

where z is the vertical coordinate. Since $\nabla\tilde{p} = \nabla p + \alpha\mathbf{e}_z = \nabla p + \alpha\mathbf{g}$, the momentum balance simplifies to

$$\nabla^2\mathbf{v} - \nabla\tilde{p} = 0, \text{ in } \Omega. \tag{22}$$

We may define a modified stress tensor $\tilde{\sigma}$ by

$$\tilde{\sigma} := \tilde{\sigma}(\tilde{p}, \mathbf{v}) = -\alpha z\mathcal{I} + \sigma(p, \mathbf{v}). \tag{23}$$

Substitution of this new stress tensor into the boundary conditions at S_0 and S_1 yields

$$\begin{aligned} \tilde{\sigma}\mathbf{n} &= -(\alpha z + \beta\kappa)\mathbf{n}, \text{ at } S_0, \\ \tilde{\sigma}\mathbf{n} &= -(1 + \alpha z + \beta\kappa)\mathbf{n}, \text{ at } S_1. \end{aligned} \tag{24}$$

It can be verified that substitution of $\tilde{\sigma}$ into the second part of the boundary condition at S_3 yields

$$\alpha z\mathbf{n}.\mathbf{t} + (\tilde{\sigma}\mathbf{n} + \beta_m\mathbf{v}).\mathbf{t} = 0. \tag{25}$$

The first term cancels out since $\mathbf{n}.\mathbf{t} = 0$ and we obtain the following conditions at S_3,

$$\begin{aligned} (\tilde{\sigma}\mathbf{n} + \beta_m\mathbf{v}).\mathbf{t} &= 0, \\ \mathbf{v}.\mathbf{n} &= 0, \text{ at } S_3, \end{aligned} \tag{26}$$

To summarize, the equations and boundary conditions in dimensionless form are given by

$$\begin{aligned} \nabla^2\mathbf{v} - \nabla p &= 0, \text{ in } \Omega, \\ \nabla.\mathbf{v} &= 0, \text{ in } \Omega, \\ \tilde{\sigma}\mathbf{n} &= -(\alpha z + \beta\kappa)\mathbf{n}, \text{ at } S_0, \\ \tilde{\sigma}\mathbf{n} &= -(1 + \alpha z + \beta\kappa)\mathbf{n}, \text{ at } S_1, \\ \mathbf{v} &= \mathbf{0}, \text{ at } S_2, \\ (\tilde{\sigma}\mathbf{n} + \beta_m\mathbf{v}).\mathbf{t} &= 0, \text{ at } S_3, \\ \mathbf{v}.\mathbf{n} &= 0, \text{ at } S_3. \end{aligned} \tag{27}$$

In the sequel we will omit the ˜ to simplify the notation.

This paper does not aim to address the heat change in the glass and the heat exchange between glass, air and mould. However, the temperature does enter the equations via the viscosity of the glass. In general the viscosity η depends on the temperature of the glass. Often in glass problems this dependance is modelled with the Vogel-Fulcher-Tamman relation [12], which is given as

$$\log_{10}\eta = A + \frac{B}{T - T_0}, \tag{28}$$

Here T is the temperature of the glass and A, B and T_0 are the so-called Lakatos coefficients [12].

We show that for the blowing problem it is not necessary to include the temperature dependance of the viscosity into our model. For this goal, we consider the energy equation,

$$\rho c_p \frac{\partial T}{\partial t} + \mathbf{v}.\nabla T \quad = \quad k_c \nabla^2 T + \nabla \left(k_r(T) \nabla T \right) + \eta \left(\left(\nabla \mathbf{v} + \nabla \mathbf{v}^T \right) : \nabla \mathbf{v} \right). \tag{29}$$

The three terms at the right-hand side of this equation represent the conduction, the radiation and the viscous dissipation, respectively. The parameters c_p, k_c and k_r are the specific heat, the conductivity and the Rosseland parameter, respectively. Introduce a dimensionless temperature T' by $T = T_m + \Delta T T'$, with $\Delta T = T_g - T_m$, T_g being the initial temperature of the glass, and T_m being the temperature of the mould. We also introduce a typical timescale t_c, $t_c := L/v_c$. The dimensionless form of the heat equation reads (we immediately omit the $'$)

$$\frac{\partial T}{\partial t} + \mathbf{v}.\nabla T \quad = \quad \frac{1}{\text{Pe}} \nabla^2 T + \nabla \left(\frac{k_r(T)}{k_c} \frac{1}{\text{Pe}} \nabla T \right) + \frac{\text{Ec}}{\text{Re}} \left(\left(\nabla \mathbf{v} + \nabla \mathbf{v}^T \right) : \nabla \mathbf{v} \right). \tag{30}$$

Here the Péclet number and the Eckert number are defined as

$$\text{Pe} = \frac{\rho c_p v_c L}{k_c}, \quad \text{Ec} = \frac{v_c^2}{c_p \Delta T}. \tag{31}$$

Typical values of the parameters involved (see Table 1) yield $1/\text{Pe} = \mathcal{O}\left(10^{-3}\right)$ and $\text{Ec}/\text{Re} = \mathcal{O}\left(10^{-7}\right)$. Hence the energy equation reduces to

$$\frac{\partial T}{\partial t} + \mathbf{v}.\nabla T = 0. \tag{32}$$

This implies that the temperature remains constant along streamlines. If we assume that the initial temperature of the glass is uniform in space and the mould has a uniform temperature too, it follows that the glass has a uniform temperature distribution. As a consequence, also the viscosity can be taken uniform. This shows that the Stokes equations and the heat equation are decoupled equations and can be solved separately. In this thesis however, we are only interested in the flow problem. Therefore we will not atempt to solve the heat equation.

3 Boundary element method

We use the boundary element method [13, 14, 15] to solve the Stokes problem outlined in the previous section. First we show how the boundary value problem transforms into a set of boundary integral equations. After discretisation of the surface we obtain a linear system of algebraic equations. Solving this system yields the velocity of the glass surface.

The key ingredient to transform the mathematical model from he previous section into a set of boundary integral equations is Green's identity for the Stokes problem. For an extensive derivation we refer the reader to [16].

parameter	value	dimension		parameter	value	dimension
L	0.01	m		T_g	1100	^{o}C
v_c	0.01	m s^{-1}		T_m	600	^{o}C
ρ	2500	kg m^{-3}		$p_0 g$	$1.00 \cdot 10^5$	kg m^{-1}s^{-2}
η	1000	kg m^{-1}s^{-1}		p_1	$1.38 \cdot 10^5$	kg m^{-1}s^{-2}
c_p	1350	J kg^{-1}K^{-1}		γ	0.3	kg s^{-2}
k_c	1.5	W m^{-1}K^{-1}				

Table 1: *Material properties and process parameters for the glass blowing problem.*

We introduce a new variable $\mathbf{b}$,

$$\mathbf{b} := \sigma(p, \mathbf{v})\mathbf{n}, \tag{33}$$

which represents the normal stress at the boundary. Under the assumption that the surface of Ω is smooth, it can be deduced that

$$\begin{aligned} \frac{1}{2}\delta_{ij}v_j(\mathbf{x}) \quad &+ \int_S q_{ij}(\mathbf{x},\mathbf{y})v_j(\mathbf{y})dS_y \\ &= \int_S u_{ij}(\mathbf{x},\mathbf{y})b_j(\mathbf{y})dS_y, \ \mathbf{x} \in S, \end{aligned} \tag{34}$$

for $i = 1, 2, 3$. Here the functions q_{ij} and u_{ij} are defined as

$$\begin{aligned} q_{ij} \quad &:= \quad \frac{3}{4\pi}\frac{(x_i - y_i)(x_j - y_j)(x_k - y_k)n_k}{\|\mathbf{x} - \mathbf{y}\|^5} \\ u_{ij} \quad &:= \quad \frac{1}{8\pi}\Big[\delta_{ij}\frac{1}{\|\mathbf{x} - \mathbf{y}\|} + \frac{(x_i - y_i)(x_j - y_j)}{\|\mathbf{x} - \mathbf{y}\|^3}\Big]. \end{aligned} \tag{35}$$

We introduce the integral operators $\mathcal{G}$ and $\mathcal{H}$,

$$\begin{aligned} (\mathcal{G}\boldsymbol{\phi})_i \quad &:= \quad \int\limits_S u_{ij}(\mathbf{x},\mathbf{y})\phi_j(\mathbf{y})dS_y, \\ (\mathcal{H}\boldsymbol{\psi})_i \quad &:= \quad \int\limits_S q_{ij}(\mathbf{x},\mathbf{y})\psi_j(\mathbf{y})dS_y. \end{aligned} \tag{36}$$

These operators are the single and double layer operator for the Stokes flow respectively. With these operators the boundary integral equation (34) is simply as,

$$\big(\frac{1}{2}\mathcal{I} + \mathcal{H}\big)\mathbf{v} = \mathcal{G}\mathbf{b}. \tag{37}$$

This boundary integral equation expresses the relation between the flow $\mathbf{v}$ of the surface of the fluid and the normal stresses $\mathbf{b}$ at the surface.

The surface S is approximated by K linear triangular elements. Each element typically consists of three nodes $\mathbf{x}^1$, $\mathbf{x}^2$, $\mathbf{x}^3$ that are located at the corners of the triangle. The total number of nodes is denoted by N. We introduce three linear shape functions,

$$\begin{aligned} \phi_1(\xi_1,\xi_2) &= 1-\xi_1-\xi_2, \\ \phi_2(\xi_1,\xi_2) &= \xi_1, \\ \phi_3(\xi_1,\xi_2) &= \xi_2, \end{aligned} \tag{38}$$

where $0 \le \xi_1, \xi_2 \le 1$ and $\xi_1+\xi_2 \le 1$. Consider the k-th element S_k with nodes $\mathbf{x}^1$, $\mathbf{x}^2$ and $\mathbf{x}^3$. The element S_k is parameterized by

$$\mathbf{y} = \mathbf{y}(\xi_1,\xi_2) = \phi_1\mathbf{x}^1 + \phi_2\mathbf{x}^2 + \phi_3\mathbf{x}^3. \tag{39}$$

The vectors $\mathbf{v}$ and $\mathbf{b}$ are linearly approximated with the same shape functions,

$$\begin{aligned} \mathbf{v}(\mathbf{y}) &= \phi_1\mathbf{v}^1 + \phi_2\mathbf{v}^2 + \phi_3\mathbf{v}^3, \\ \mathbf{b}(\mathbf{y}) &= \phi_1\mathbf{b}^1 + \phi_2\mathbf{b}^2 + \phi_3\mathbf{b}^3. \end{aligned} \tag{40}$$

Here $\mathbf{v}^s = \mathbf{v}(\mathbf{x}^s)$ is the velocity at the node $\mathbf{x}^s$ and $\mathbf{b}^s = \mathbf{b}(\mathbf{x}^s)$ is the normal stress at the node $\mathbf{x}^s$. We approximate the surface integral over S in (34) by a sum of integrals over the elements S_k, and substitute the approximations for $\mathbf{v}$ and $\mathbf{b}$,

$$\begin{aligned} \frac{1}{2}v_i(\mathbf{x}) &+ \sum_{k=1}^{K}\int_{S_k} q_{ij}(\mathbf{x},\mathbf{y})\Big(\phi_1 v_j^1 + \phi_2 v_j^2 + \phi_3 v_j^3\Big)dS_y \\ &= \sum_{k=1}^{K}\int_{S_k} u_{ij}(\mathbf{x},\mathbf{y})\Big(\phi_1 b_j^1 + \phi_2 b_j^2 + \phi_3 b_j^3\Big)dS_y, \\ &\quad \mathbf{x}\in S,\ i=1,2,3. \end{aligned} \tag{41}$$

We substitute $\mathbf{x} = \mathbf{x}^p$, $p = 1,\ldots,N$, in (41), obtaining $3N$ equations. Next we construct two coefficient vectors,

$$\begin{aligned} \mathbf{v} &= \left[v_1^1, v_2^1, v_3^1, \ldots, v_1^N, v_2^N, v_3^N\right]^T, \\ \mathbf{b} &= \left[b_1^1, b_2^1, b_3^1, \ldots, b_1^N, b_2^N, b_3^N\right]^T. \end{aligned} \tag{42}$$

This allows us to write (41) in a matrix-vector form,

$$\mathrm{H}\mathbf{v} = \mathrm{G}\mathbf{b}. \tag{43}$$

To compute the matrices H and G, we have to evaluate integrals of the form

$$\int_{S_k} q_{ij}(\mathbf{x}^p,\mathbf{y})\phi_r dS_y, \quad \int_{S_k} u_{ij}(\mathbf{x}^p,\mathbf{y})\phi_r dS_y. \tag{44}$$

The integrals can be evaluated by using a Gauss quadrature scheme, but special care has to be taken when the node $\mathbf{x}^p$ is in the surface element S_k. In that case one need to use a slightly more elaborate method to evaluate the integrals, e.g. a logarithmic Gauss quadrature scheme.

In the case where $S_3 = \emptyset$ we either know the velocity coefficients at a node or the normal stress coefficients. Hence in (43) some of the unknowns are in the vector $\mathbf{b}$ at the right-hand side and some of the knowns are in the vector $\mathbf{v}$ at the left-hand side. By interchanging columns properly we arrive at the standard form linear system with unknown $\mathbf{x}$,

$$\mathrm{A}\mathbf{x} = \mathbf{f}. \tag{45}$$

When $S_3 \neq \emptyset$ there are nodes at which both the velocity and normal stress coefficients are unknown, though related via the slip conditions (26). Let $\mathbf{t}^r$, $r = 1, 2$, be the two tangential vectors at the wall at such a node $\mathbf{x} \in S_3$. Since $\mathbf{v}.\mathbf{n} = 0$ at $\mathbf{x}$, we may write

$$\mathbf{v}(\mathbf{x}) = a_1\mathbf{t}^1(\mathbf{x}) + a_2\mathbf{t}^2(\mathbf{x}),\ a_1, a_2 \in \mathbf{R}. \tag{46}$$

Substitution into $(\mathbf{b} + \beta_m\mathbf{v}).\mathbf{t}^r = 0$ yields $a_r = -(\mathbf{b}.\mathbf{t}^r)/\beta_m$. In the boundary integral equation we replace $\mathbf{v}(\mathbf{x})$ by the above expression. Thus we have eliminated $\mathbf{v}(\mathbf{x})$ and the only uknown at $\mathbf{x}$ is $\mathbf{b}(\mathbf{x})$. The solution of the BEM yields the normal stress $\mathbf{b}(\mathbf{x})$ and as a post-processing step we compute the velocity $\mathbf{v}(\mathbf{x})$ from (46). In this way we arrive at the same standard form linear system $\mathrm{A}\mathbf{x} = \mathbf{f}$.

The matrix A is a dense matrix and the linear system can be solved by using an LU-decomposition technique. Due to the dense nature of the matrix, this may become costly, especially when the size of the matrix is large.

4 Algorithm

In this section we describe an algorithm to simulate the blowing phase. Several steps can be distinguished in this algorithm.

```
Initial surface $S$
for step $= 1, 2, ...$
   Use BEM to obtain $\mathbf{v}$
   Perform velocity smoothing
   Perform time integration to update $S$
   Perform Laplacian smoothing
   Regridding
end
```

Time integration

The movement of the surface of the fluid domain is described by the velocity field $\mathbf{v}(\mathbf{x}, t)$ that is the outcome of the Stokes problem. In fact we calculate the velocity at a set of N nodes at the surface. To study the evolution of the surface we need to solve an ordinary differential equation,

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{v}(\mathbf{x}, t),\ \mathbf{x} \in S. \tag{47}$$

Assume that at time $t = t^n$ we know the locations of the nodes $\mathbf{x}^n$ and the velocity at these nodes $\mathbf{v}(\mathbf{x}^n, t^n) =: \mathbf{v}^n$. We do not have any information of the nodes or velocity in the future. Therefore we cannot make use of implicite time integration schemes to solve (47).

An option is to use an *Euler forward* scheme, in which we approximate the locations of the nodes at the next time level t^{n+1} by

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}(\mathbf{x}^n, t^n). \tag{48}$$

However this scheme is only first order accurate. Another option is to use a modified version of *Heun's method*, which is also called the *improved Euler method*. This method is known to be second order accurate [17]. However for this method we need the velocity $\mathbf{v}$ at the next time level t^{n+1} in the new location $\mathbf{x}^{n+1}$ of the node. As we remarked before we do not have information of future time levels. To get around this problem we first predict the location of the node at the next time level using a Euler forward step (48). For this predicted node $\mathbf{x}^{n+1}$ we again solve the Stokes problem and we obtain the velocity in this node at time t^{n+1}. Then we update our prediction of $\mathbf{x}^{n+1}$ with Heun's method. In this way we corrected the prediction of $\mathbf{x}^{n+1}$ as performed with the Euler forward step. The disadvantage of the Heun's method is that we have to solve two Stokes problems at each time step.

A third option to perform time integration is the so called *flow method* developed in [18]. Time integration is explicit in this method and only one Stokes problem needs to be solved at each time step, while accuracy is second order. To reach this quadratic accuracy an inverse interpolation problem is solved at each time step. The method exploits the fact that the time-dependence of the velocity is very small, $\partial \mathbf{v}/\partial t \approx 0$, and hence we have to solve an ODE of the form $\partial \mathbf{x}/\partial t = \mathbf{v}(\mathbf{x})$. Another advantage of the flow method is its volume-conserving nature. Also on the long term this method performs better than other second order methods. However, the time interval that is spanned in our simulations is very restricted, so we cannot exploit the long term performance of the flow method.

Note that the BEM with linear elements as described in the previous section is second order accurate. This means that we cannot improve the overall accuracy by choosing an accurate time integration method only. We also need to use higher order elements in the BEM to achieve high accuracy. To illustrate this, we may monitor the total volume of the fluid domain. As we are dealing with an incompressible fluid, the volume should remain constant during the blowing phase. Hence computation of the fluid volume at each time step provides us insight in the accuracy of the simulation tool.

At each node at the surface the BEM computes the velocity with an error of order h^2, where h is a typical size of the boundary elements. Summation over all elements yields an error of order h in the total volume. From that point of view it does not make any difference if we use a first or second order accurate time integration scheme. The error in the fluid volume is always dominated by the error made by the BEM. Hence in our simulation we always choose the Euler forward method to perform time integration.

Laplacian smoothing

A well-known technique to smooth a triangulated surface is *Laplacian smoothing* [19, 20, 21]. For each node $\mathbf{x}$ at the surface S we compute the geometric average $\mathbf{x}_{av}$ of the neighbouring nodes. A neighbouring node is a node that shares an edge of a triangle with $\mathbf{x}$. If the node $\mathbf{x}$ is too far away from $\mathbf{x}_{av}$, it is relocated at the geometric average. Or more general, the node $\mathbf{x}$ is replaced by a weighted average of $\mathbf{x}$ and $\mathbf{x}_{av}$,

$$\mathbf{x} \rightarrow (1-w)\mathbf{x} + w\mathbf{x}_{av}, \tag{49}$$

where w is a suitably chosen weight. We can do this for every node, in which case we apply *global smoothing*. Or we may replace $\mathbf{x}$ only if the distance to $\mathbf{x}_{av}$ exceeds a certain tolerance. In that case we apply *local smoothing*. In other words, we smooth the surface only at nodes where it is most needed. The process can be repeated several times. In each iteration the surface gets smoother.

A side-effect of the smoothing is that the volume that the surface encloses decreases. This is a typical disadvantage of standard Laplacian smoothing. There are several modifications to the standard technique to avoid volume loss. The simplest one is to restrict the movement of the node $\mathbf{x}$ to a direction perpendicular to the normal at the surface at $\mathbf{x}$. Unfortunately this reduces the performance of the smoothing. Another possibility is to take pairs of nodes that are connected by an edge. The two nodes are relocated to new positions simultaneously. In this way we have more freedom to move the nodes to the desired locations, while conserving the volume. In our simulation tool we use the latter modification to Laplacian smoothing. For more details we refer to [20].

Regridding

As the deformation of the glass is large the triangular elements of the discretised surface S may become very large. Therefore it is necessary to remesh the surface regularly. At every time step of the simulation we monitor the length of the egdes of the boundary elements. If such an edge has a length larger than a certain tolerance level, this edge is subdivided. As a consequence the two elements that share this edge are subdivided into four new elements.

5 Results

In this section we perform several simulations with model glass parisons that are blown to bottles or jars. In all simulations we assume that the top of the parison is fixed, i.e. the velocity of the glass is equal to zero. This part of the glass corresponds to the surface part S_2 (see Figure 2).

The material properties of glass and the process parameters can be found in Table 1. With these properties the dimensionless parameters that appear in the model get the following values,

$$\alpha = 0.006, \quad \beta = 0.001. \tag{50}$$

Ideally, the value of the dimensionless friction coefficient β_m has to be determined experimentally. However, to the authors' knowledge no such experiments have been reported in literature. For our simulations we take $\beta_m = 1$, i.e. a partial-slip condition for the glass when it comes into contact with the wall.

The first simulation shown in Figures 3 and 4 concerns a glass parison that is put into a cylindrical mould. This example can also be seen as a movie at the journal's website. The pressure that is blowing in from above causes the glass to move in vertical and radial directions. After some time the whole mould is filled with glass, that is the walls of the mould are covered with a layer of glass, see Figure 4. The glass is allowed to slip along the wall when it comes into contact with the mould. It turns out that it is very hard to make the glass fill the corners of the mould. To improve the filling of the corners we perform local mesh refinement.

The cross-sectional view in Figure 4 shows that we get sharp corners at the top of the parison where the surface parts S_1 and S_2 touch. This is a direct consequence of the choice to keep the glass fixed at S_2, while it is allowed to move at S_1. In reality these sharp corners do not appear.

One can also see that, although we perform local mesh refinement, still some small gaps appear between the glass and the mould. Such gaps will always be present in our simulations as we try to fill a smoothly curved mould by a set of straight linear elements.

Figure 5 and 6 show a simulation with a mould that has a more challenging shape, though still rotationally symmetric. The lower part of the mould has a smaller width than the upper part of the mould. Again the corners are rounded. The glass is allowed to slip along the wall of the mould. This example can also be seen as a movie at the journal's website. In the previous simulation the only driving force was the pressure of the air that is blown into the parison from above. In reality the parison is first subjected to gravity only. The glass will sag to the bottom of the mould and when it almost touches the bottom, air starts to blow into the mould. The simulation in Figure 5 and 6 consists of these two stages. The first three snapshots correspond to the sagging stage while the last three snapshots correspond to the blowing stage. It is observed that during the sagging the glass mainly moves in vertical direction. During the blowing the glass both moves in vertical and radial direction. It turns out that gravity has little effect during the blowing stage. Therefore the gravity coefficient α can be set equal to zero as soon as the blowing stage starts.

The simulation that is presented in Figure 7 and 8 deals with a parison and a mould that are not rotationally symmetric. The initial parison has the shape of a box with rounded corners. Also the mould has the shape of a box, but the upper part has a smaller width than the lower part. Again all corners are rounded. This example can also be seen as a movie at the journal's website. Although the parison and mould are symmetric in the planes $x = 0$ and $y = 0$, these symmetries are not exploited in the computations. Again we consider the two stages that occur in the production process: sagging and blowing. The first three snapshots correspond to the sagging stage while the last three snapshots correspond to the blowing phase. In this case it is clearly visible that the glass moves in vertical direction only during the sagging stage. It is only in the blowing stage that the glass also moves in the radial direction.

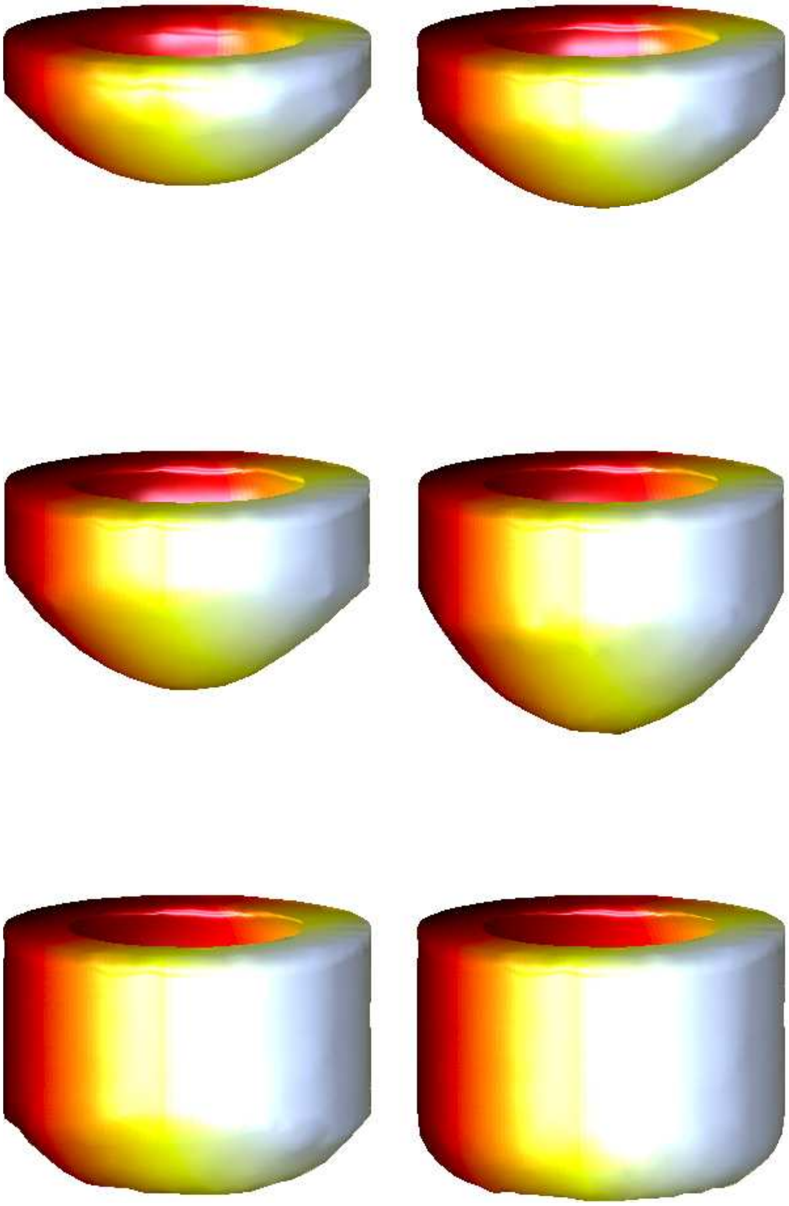


Figure 3: *3D Snapshots of the glass as it expands due to the pressure blowing in from above. The mould has a cylindrical shape with rouned corners. The glass is allowed to slip along the wall.*

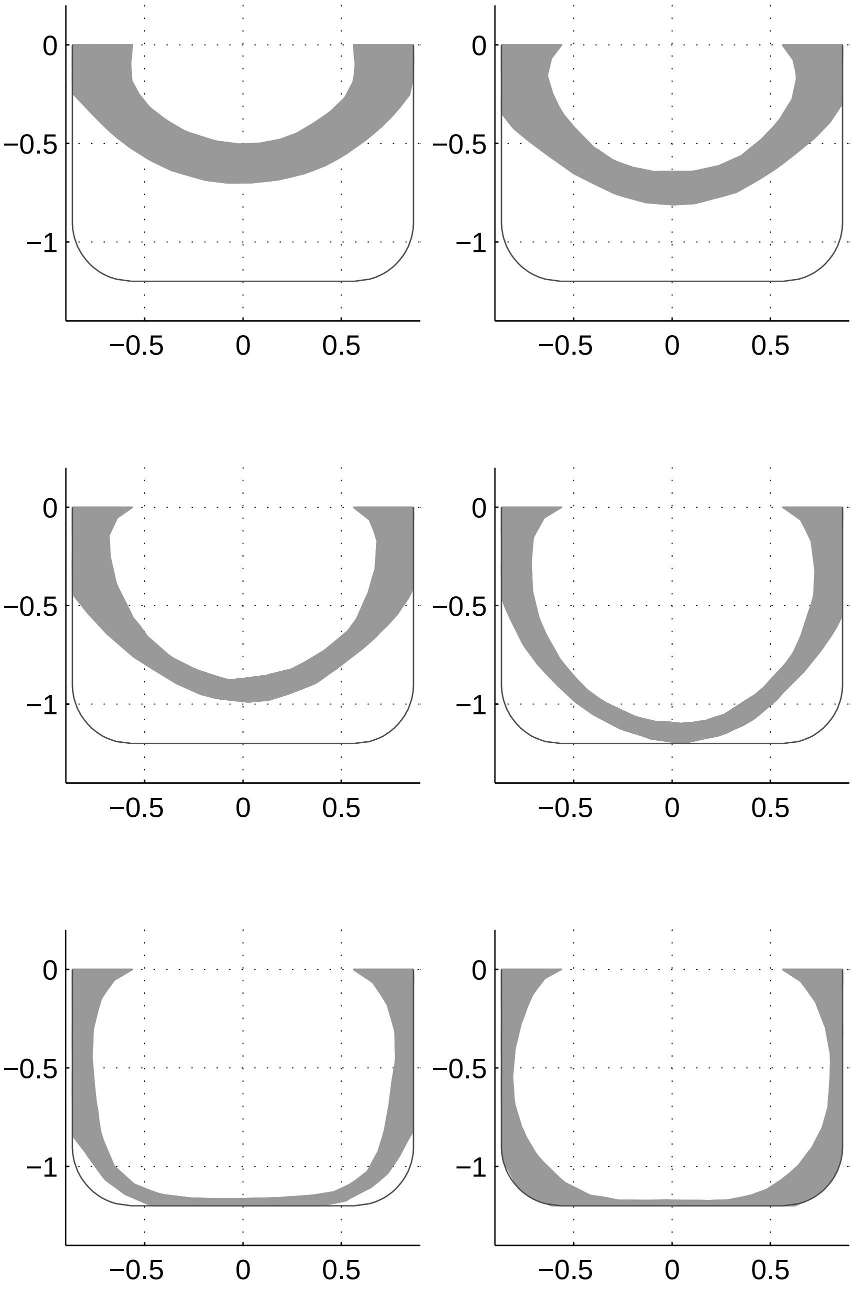


Figure 4: *Cross-sectional view of Figure 3 at $y = 0$.*

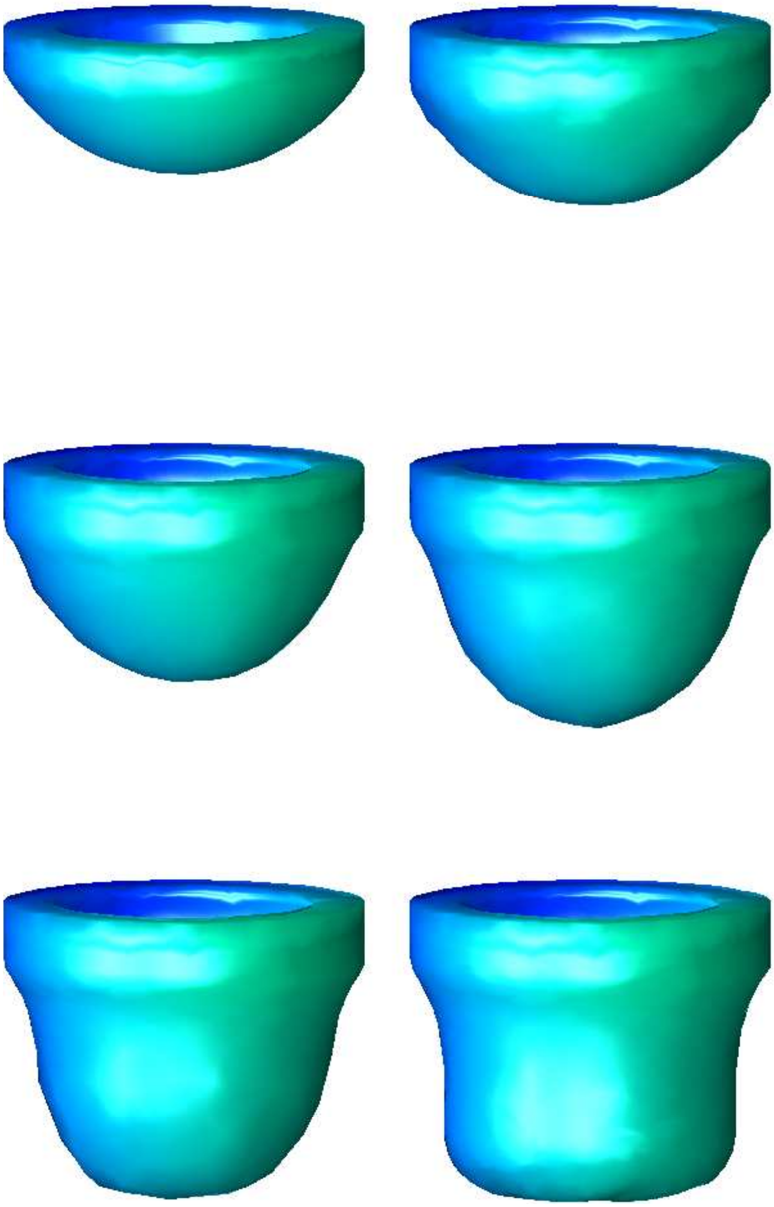


Figure 5: *During the first three snapshots the glass is sagging to the bottom of the mould. During the last three snapshots air is blowing into the mould from above. The glass is allowed to slip along the wall of the moud.*

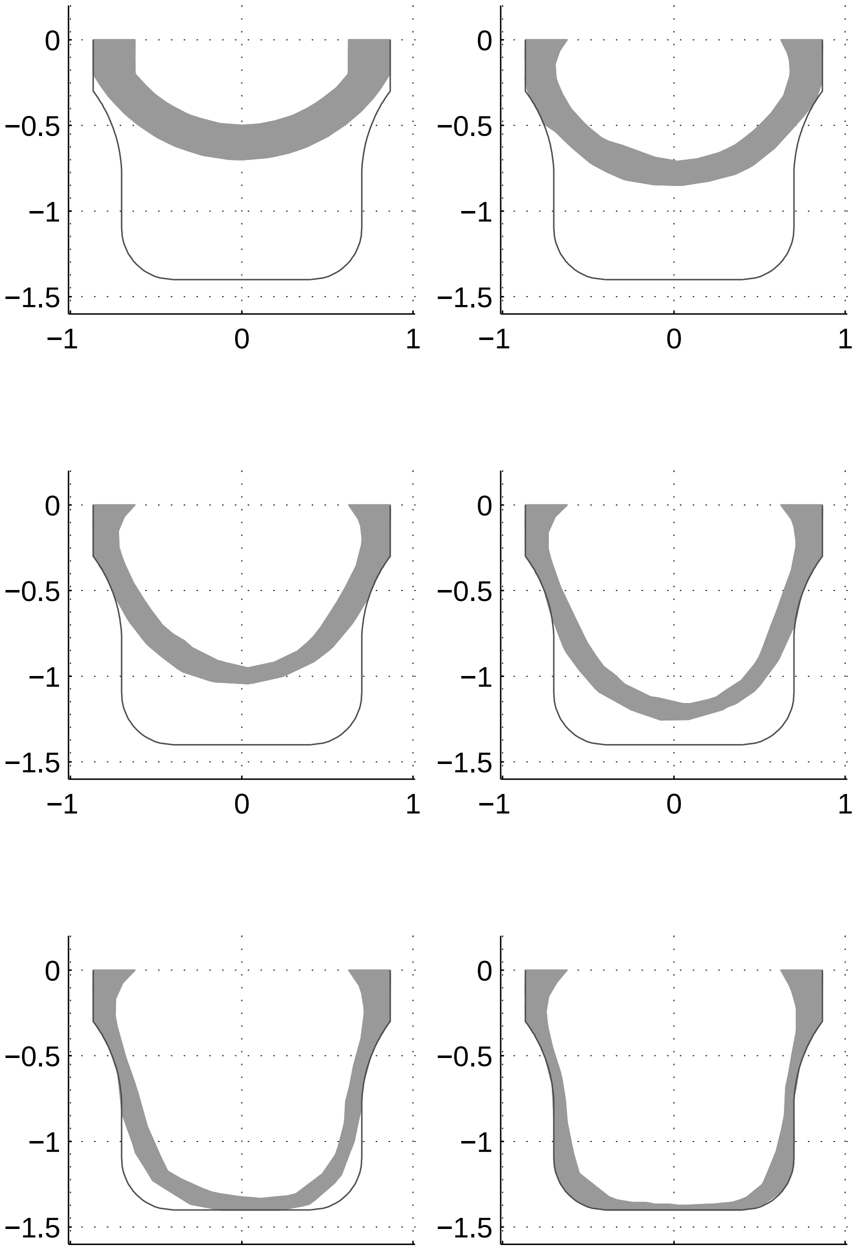


Figure 6: *Cross-sectional view of Figure 5 at $y = 0$.*

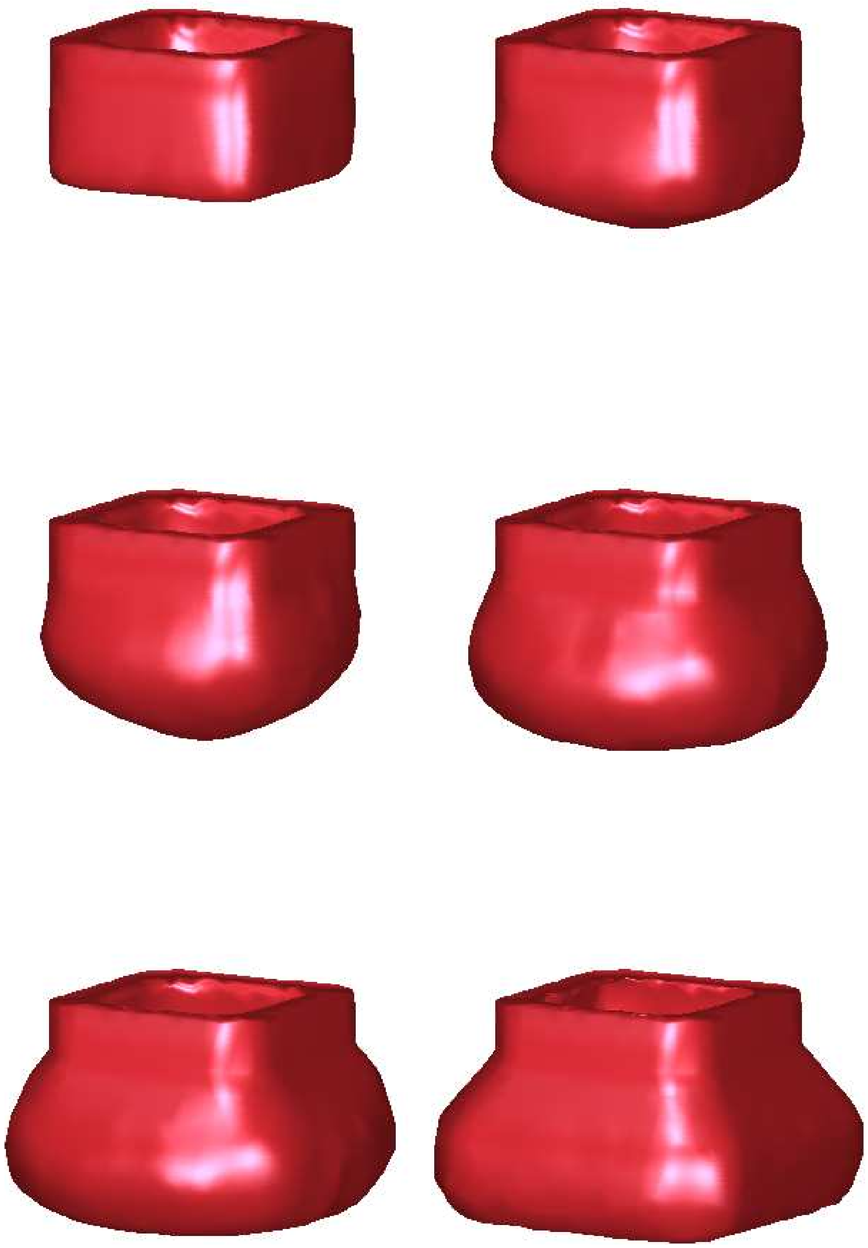


Figure 7: *A square parison with rounded corners and a square mould with rounded corners. During the first three snapshots the glass is sagging to the bottom of the mould. During the last three snapshots air is blowing into the mould from above. The glass is not allowed to slip along the wall.*

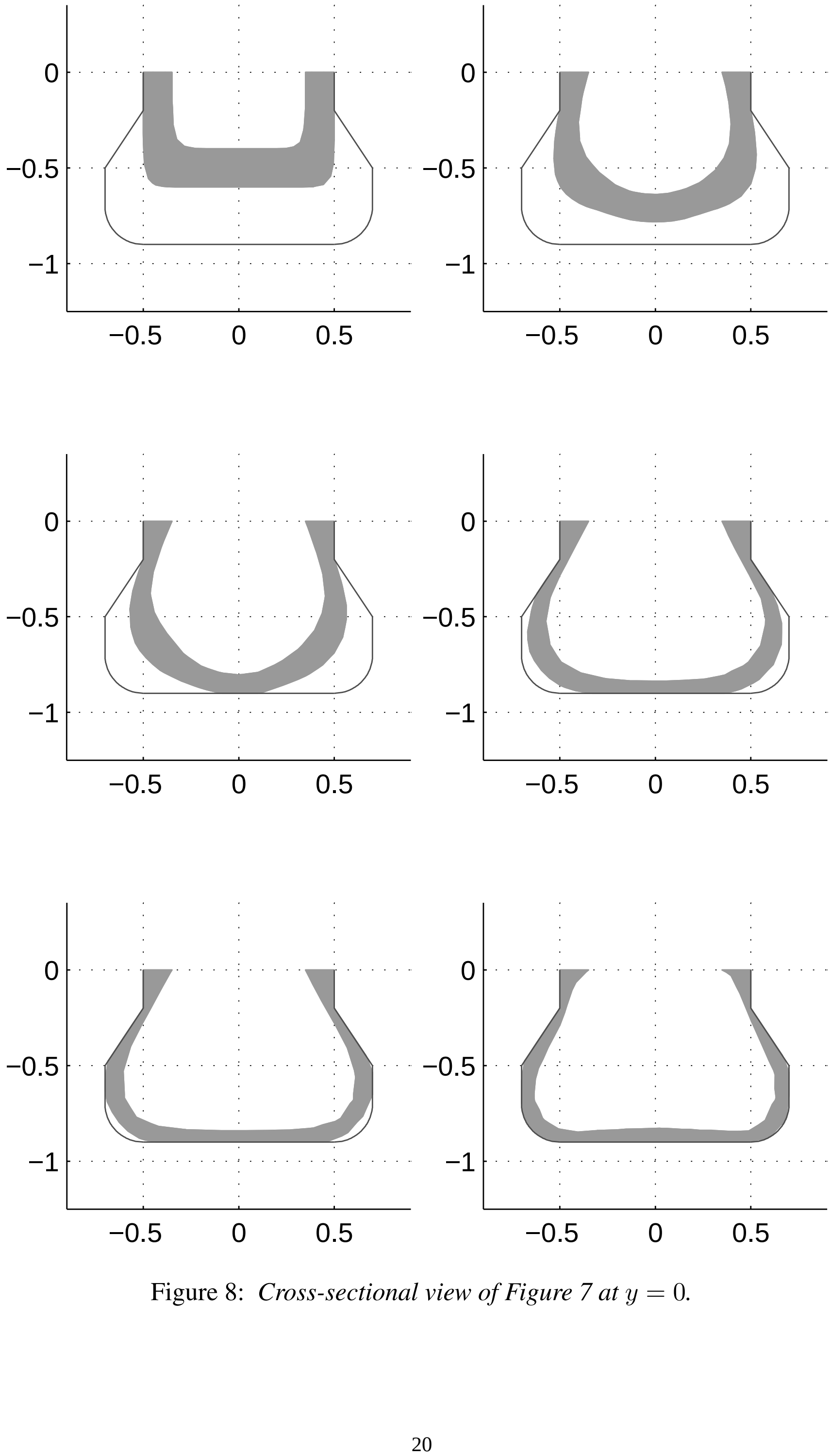


Figure 8: *Cross-sectional view of Figure 7 at $y = 0$.*

6 Discussion

We have developed a simulation tool to analyse the blowing phase in the production process of glass containers. All calculations are performed in three dimensions, which allows to study parisons that are not rotationally symmetric. For instance, one can see how a certain imperfection in the initial parison develops through time. In principle our simulation tool can handle complex shapes of the parison and mould. However, the shape of the mould has to be described mathematically, and such a description might be very difficult to realize for complex shapes.

The boundary element method requires the existence of a fundamental solution for the boundary value problem. For the Stokes equations such a fundamental solution can be found, provided that the coefficients in the Stokes equations are constant. However, for hot liquid glass the material parameters are known to be temperature-dependent, in particular the viscosity. As the temperature is time and space dependent, so is the viscosity. Hence in reality the coefficients in the Stokes equations are not constant and a fundamental solution is not known. Dimensional analysis of the energy equation shows that the temperature dependance of the viscosity is negligible, and therefore we assume that the viscosity is constant.

Little is known about the friction parameter β_m. To the authors' knowledge there are no experiments mentioned in literature in which the friction parameter for glass is determined. For the application studied in this paper it is known that there is little friction between glass and mould. Therefore, a small value of β_m seems to be an appropriate choice.

Another advantage of the boundary element method is the relative ease with which surface tension can be added to the model and incorporated in the computations. During the blowing phase this surface tension does not have much influence, but during the sagging of the glass it cannot be neglected.

The boundary element method is an appropriate numerical method for simulating blowing problems. As we are only interested in the shape evolution, i.e. the flow of the glass surface, it is very efficient to use the boundary element method, since it only discretises the surface of the glass. Many other numerical method also require the computation of the flow in the interior of the glass. As a direct consequence the matrices that appear in the boundary element method are much smaller than the matrices that appear in the finite element method, for instance. To compute the flow of the glass during the blowing phase, the boundary element method requires a computation time ranging from half an hour to an hour. This is reasonably fast keeping in mind the complex nature of the equations at hand.

Although we focused on the blowing problem for glass products, the developed simulation tool can also be used for other types of blowing problems. The current model is valid for viscous fluids, but can be modified to, for instance, visco-elastic fluids. We believe that the boundary element method will prove to be an efficient numerical method for simulating this larger class of blowing problems.

References

[1] C. Marechal, P. Moreau AND, D. Lochegnies, Numerical optimization of a new robotized glass blowing process, Engineering with Computers, 19: 233–240, 2004

[2] J.M.A. César de Sá, Numerical modelling of glass forming processes, Eng. Comput., 3: 266–275, 1986

[3] J.M.A. César de Sá, R.M. Natal Jorge, C.M.C. Silva, R.P.R. Cardoso, in, A computational model for glass container forming processes, Europe Conference on Computational Mechanics Solids, Structures and Coupled Problems in Engineering, 1999

[4] C.G. Giannopapa, Level-set method used to track the glass-air interface in the blow step of glass containers, ASME J. Manuf. Sci. Eng., in press, 2007

[5] J.A. Sethian, J.D. Strain, Crystal growth and dendritic solidification, J. Comput. Phys., 98: 231–253, 1992

[6] Z. Li, W. Cai, A level set-boundary element method for simulation of dynamic powder consolidation of metals, in Numerical Analysis and its Applications 2000, Lecture Notes in Computer Science 1988, editors L. Vulkov AND J. Was̀iewski AND P. Yalamov, 527–534, 2001

[7] M. Garzon, D. Adalsteinsson, L. Gray, J.A. Sethian, A coupled level set-boundary integral method for moving boundary simulations, Interfaces Free Bound., 7: 277–302, 2005

[8] G.A.L. Vorst van der, R.M.M. Mattheij, H.K. Kuiken, Boundary element solution for two-dimensional viscous sintering, J. Comput. Phys., 100: 50–63, 1992

[9] V. John, A. Liakos, Time-dependent flow across a step: the slip with friction boundary condition, Int. J. Numer. Meth. Fluids, 50: 713–731, 2006

[10] U. Lange, The use of remeshing methods in pressing simulations, in, Mathematical Simulation in Glass Technology, editors D. Krause and H. Loch, Springer, 317–325, 2002

[11] C. Pozrikidis, A practical guide to boundary element methods, Chapman and Hall, Boca Raton, 2002

[12] R.C.G. Beerkens and others, NCNG Handbook for Glass Technologists, 1997.

[13] A.A. Becker, The Boundary Element Method in Engineering, McGraw-Hill, Maidenhead, 1992

[14] C.A. Brebbia, J. Dominguez, Boundary elements: an introductory course, McGraw-Hill London, 1989

[15] C.A. Brebbia, J.C.F. Telles, L.C. Wrobel, Boundary Element Techniques: Theory and Applications in Engineering, Springer-Verlag, Berlin, 1984

[16] O.A. Ladyzhenskaya, The Mathematical Theory of Viscous Incompressible Flow, Gordon and Beach, New York-London, 1963

[17] J.C. Butcher, The numerical analysis of ordinary differential equations, Wiley, Chichester, 1987

[18] B. Tasić, R.M.M. Mattheij, Explicitly solving vectorial ODEs, Appl. Math. Comput., 164: 913–933, 2005

[19] D. Field, Laplacian smoothing and Delaunay triangulations, Comm. Appl. Numer. Methods, 4(6): 709–712, 1988

[20] A. Kuprat, A. Khamayseh, D. George, L. Larkey, Volume conserving smoothing for piecewise linear curves, surfaces and triple lines, J. of Comp. Phys., 172: 99–118, 2001

[21] J. Vollmer, R. Mencl, H. Müller, Improved Laplacian smoothing of noisy surface meshes, Computer Graphics Forum, 18(3): 131–138, 1999